Auto-Parser: Android Auto and Apple CarPlay Forensics

Andrew Mahr(✉), Robert Serafin, Cinthya Grajeda, and Ibrahim Baggili

Connecticut Institute of Technology, University of New Haven Cyber Forensics Research and Education Group (UNHcFREG), West Haven, USA
amahr1@unh.newhaven.edu, {rsera1,cgrajedamendez,ibaggili}@newhaven.edu

Abstract. Mobile device features like Apple CarPlay and Android Auto provide drivers safer hands-free navigation methods to use while driving. In crash investigations, understanding how these applications store data may be crucial in determining the what, when, where, who and why. By analyzing digital artifacts generated by Android Auto and Apple CarPlay, investigators can determine the last application displayed on the head unit, the application layout of the user's home display screen, and other evidence which points to the utilization of the mobile device and its features while driving. Additionally, usage data can be found within other applications compatible with Android Auto and Apple CarPlay. In this paper, we explore the digital evidence produced by these applications and propose a proof of concept open source tool to assist investigators in automatically extracting relevant artifacts from Android Auto and Apple CarPlay as well as other day-to-day essential applications.

Keywords: Android auto · Apple CarPlay · Mobile forensics · Artifacts

1 Introduction

Digital evidence acquired from mobile devices supporting vehicular hands-free navigation and communication may prove useful in crash investigations. According to the Centers for Disease Control and Prevention (CDC), a distraction while driving could simply be using one's cell phone to send a text message or place a call, etc. Unfortunately, in 2019, texting or emailing while driving was a commonality seen more in older teens than younger teens. The consequences of driving while being distracted could be fatal. In fact, in 2018, the CDC reported that over 2,800 people died as a result of distracted driving [7], while in 2019, this number increased by 9.9% [1]. The advancement, and increased consumption of mobile and vehicular technology is growing exponentially [9,14,34]. This caused the need for safer and tighter integration between mobile devices and cars.

Apple's CarPlay[1] was introduced in 2014 and is native to iOS. Once enabled by the user, it can only be paired to the vehicle via a USB connection or

[1] https://www.apple.com/ios/carplay/.

P. Gladyshev et al. (Eds.): ICDF2C 2021, LNICST 441, pp. 52–71, 2022.
https://doi.org/10.1007/978-3-031-06365-7_4

Bluetooth, however, only a few cars support the Bluetooth feature [31]. On the other hand, Android Auto[2] is a non-native application that was created in 2015 and can be downloaded from the Google Play store. Android Auto must also be paired via USB and Bluetooth. Since the development of these mobile applications, car manufacturers have been working to integrate them into their frameworks. Today, over 600 car models offer support for CarPlay[3], while close to 60 manufacturers offer or will soon offer support for Android Auto [3].

Consequently, our work provides a forensic analysis of the Android Auto and Apple CarPlay applications. We also present an open source Python tool, *Auto-Parser*, to extract relevant data from digital artifacts and present them in an HTML report. To the best of our knowledge, this tool is the first of its kind. *Auto-Parser* may be added as part of an investigator's forensic workflow to shed light on actions taken when using these services and applications while driving.

Our work provides the following contributions:

- A primary review of the forensic disk analysis of Apple CarPlay and Android Auto applications.
- A collection of Android Auto and Apple CarPlay digital forensic artifacts publicly shared on the Artifact Genome Project[4] [15].
- An open-source tool for the automated retrieval, analysis, and reporting of relevant digital artifacts acquired from forensic images of the mobile devices.[5]

The remainder of this paper includes related work found in Sect. 2. Section 3 discusses the approach and methodology for this work. Section 4 outlines the experimental results, while Sect. 5 explains the creation and usage of the parsing tool. Lastly, Sect. 6 concludes the paper and presents future work.

2 Related Work

At the time of writing, no peer reviewed work existed on the forensic analysis of Android Auto and Apple CarPlay applications. Moreover, to the best of our knowledge, there has not been an open source tool developed to address the type of evidence that could be collected from these applications.

Nevertheless, some blogs and presentations explored this topic [12,19–21,29]. Our work was initially motivated by posts made by Joshua Hickman on his personal blog "The Binary Hick" [19,21]. We expand on past work and provide a first of its kind open source tool to automate the forensic analysis process and present relevant data in a report. In the next sections, we highlight related work on vehicular infotainment systems, and small scale digital device forensics.

[2] https://www.android.com/auto/.
[3] https://www.apple.com/ios/carplay/.
[4] https://agp.newhaven.edu.
[5] The tool may be downloaded from https://github.com/unhcfreg/Auto-Parser-Android-Auto-Apple-CarPlay.

2.1 Vehicular Forensics

Vehicle infotainment systems may offer digital evidence related to driving activities, phone calls, and mobile messaging [26]. Past work provided a structure of embedded vehicle systems and discussed the long-term information that could be found and recovered to better understand the system's end users [26].

Other work explored the challenges of Vehicle Ad-Hoc Networks and the forensic potential of entertainment systems were discussed [27]. The authors then captured forensic images of a Ford F-150 vehicle's infotainment system and described the data that could be recovered. Another case study noted the challenges in conducting vehicle forensics and contributed a case study on a Volkswagen Golf [24]. It also important to note that the Berla Corporation specializes in the development of tools and techniques for vehicle infotainment forensics[6].

2.2 Mobile and Application Forensics

The field of small scale digital device forensics is vast. There have been numerous studies that forensically explored applications and devices. Since Android Auto and Apple CarPlay interact with mobile devices and their applications (such as WhatsApp), prior work in this domain is of relevance.

Past work examined the data stored in vehicle assistant applications for notable car manufacturers and discussed digital artifacts that could be extracted [11]. Relevant to our work is also an extensive study related to mobile GPS applications. The work was conducted on both Android and iOS mapping applications such as Waze, MapQuest and Google Maps. A tool was created based on the findings of the study to parse data found in the applications [32]. Likewise, [18] provided an analysis and algorithm for the automated retrieval of data from health and fitness applications. Their work discovered health information, passwords, geo-location data, and other private information that could be useful in an investigation.

Let us not forget that car systems also interact with mobile messaging applications. Relevant work explored the forensics of social-messaging applications such as WhatsApp, Tango, Viber, and ooVoo to determine their security and forensic footprints. This work concluded that Personally Identifiable Information (PII) could be found within application data folders as well as on servers publicly accessible [35]. Similar work and results were also conducted recently on the Zoom application concluding that PII such as chat messages, contacts, profile images, and passwords could be recovered from devices [30].

Other work purely focused on network traffic analysis of the WhatsApp call signaling protocol by decrypting its associated network traffic. The researchers uncovered the codec used for call transfer. Metadata related to the call such as phone numbers and duration can also be forensically found [25].

In a different domain of device forensics, [8] investigated the DJI Phantom III Drone and the research resulted in a tool which could recover flight data

[6] https://berla.co/.

taken from the suspect device. This work provided insight into GPS locations, speeds, flight time, and other critical information that could be recovered and used by investigators.

Additional research on social media and mobile applications on iPhones and Androids has been conducted [2,4,5,22,37]. Similarly, forensic analysis of Android vault applications was conducted [38]. Other relevant work explored the forensics of IoT devices [10], gaming consoles [33], tablets [23] and Virtual Reality headsets [6,36].

3 Methodology

We conducted our research in four main phases: scenario/case creation, data acquisition, data analysis, and the creation of a tool to parse, extract and report on all relevant artifacts stored in an acquired forensic disk image for use in an investigation. Table 1 outlines the apparatus used in our methodology. Details of these four phases and results are elaborated on.

3.1 Setup and Scenario Creation

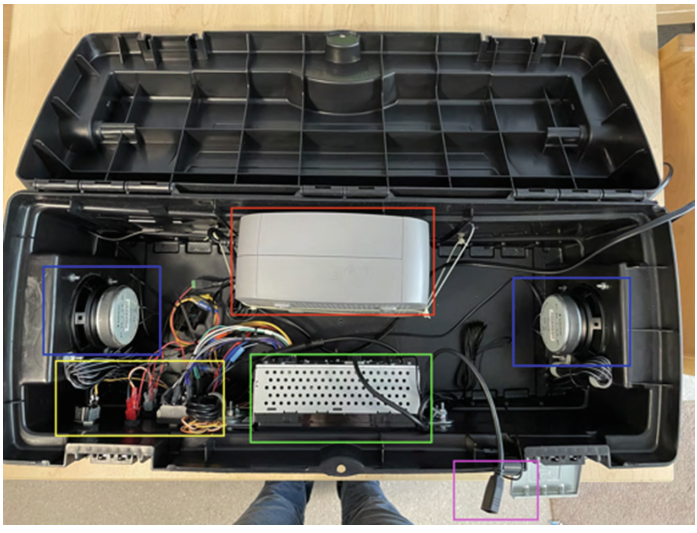


Fig. 1. Alpine stereo internal setup (Key - Red: Belkin Battery Supply, Blue: Rockford Fosgate R14X2 Speakers, Green: Alpine iLX-W650 Head Unit, Pink: USB Cable for Device Connection, Yellow: Wiring for power, speaker channels, etc.) (Color figure online)

In this phase, a stereo head unit was constructed (Figs. 1 and 2) to conduct testing in a controlled environment. At the time, a vehicle that supported these features was not available to the researchers. For some tests, this engineered,

Fig. 2. Front of alpine unit

Table 1. Apparatus

Hardware/Software	Use	Version
Galaxy S6	Android Auto	Nougat 7.0
iPhone 5s	Apple CarPlay	iOS 12.4.5, 12.4.6
Android Auto	Android Auto	5.1, 5.4, 6.0
Alpine ILX-W650	Simulated Car Stereo Unit	V1.014_1206
BELKIN Battery Backup Unit	Battery Supply	N/A
(2) Rockford Fosgate R14X2	Stereo Speakers	N/A
2020 Subaru Crosstrek	Real World Tests	Rel_UA.19.01.70
Magnet Acquire	Full Image Creator (Android/iOS)	2.25.0.20236
Autopsy	Full Image Viewer	4.17.0
Android Debug Bridge (ADB)	Android Data Extraction Tool	1.0.41, Version 29.0.6-6198805
DB Browser for SQLite	View Sqlite/ DB files	3.11.2
Xcode	View pList & XML Files	12.5 (12E262)
iBackup Viewer	pList & XML Files	4.1583
Hex Fiend	Hex Editor	Version 2.14 (1613443925)
checkra1n	iOS Jailbreak Tool	0.9.7 BETA
SuperSU	Android Jailbreak Tool	V2.82
Filza File Manager	File System Manager	3.7 Build 7

portable, stereo unit was placed inside a vehicle to simulate a real world experience. Specifically, an Alpine[7] stereo head unit was used to serve as the interface for the Android Auto and Apple CarPlay applications. The backup battery supply and speaker units served to power the device and simulate normal car interaction (Fig. 1). Once a compatible vehicle was available, the tests were also conducted with a 2020 Subaru Crosstrek Premium[8] to note any forensic differences between devices.

[7] https://www.alpine-usa.com/.
[8] https://www.subaru.com/.

To control the experiment, and obtain the most amount of data possible, communication was kept between two rooted mobile phones (Table 1). In the case of the iPhone, it was only connected via USB cable as neither head units supported Apple CarPlay connection via Bluetooth. It is important to note that Bluetooth for Apple CarPlay is supported on other vehicles. On the other hand, the Android device had to be connected via both, Bluetooth, and a USB cable for the Android Auto application to function with the head units. Additionally, to mimic other possible distractions like listening to music, the Spotify application was installed on both mobile devices. It is important to note that during the course of this research, the Android Auto application versions changed and updating the application was necessary to continue testing.

In order to simulate normal user actions with these systems, and create the most relevant data possible, the experiments listed below were conducted with both the custom built stereo head unit, and the Subaru's stereo head unit.

- Sent and received text messages between testing devices.
- Sent and received phone calls between testing devices.
- Asked Siri (iOS) and Google Assistant (Android) for driving directions to an specific address.
- Asked Siri (iOS) and Google Assistant (Android) to place phone calls and send text messages.
- Played music through the Spotify music application and interacted with the application's features on the head units.

3.2 Data Acquisition

In this phase, data was acquired from the two different mobile devices. Magnet Acquire[9] was used to obtain forensic images of each device. For the rooted Samsung, initially a full forensic image was taken, resulting in the acquisition of a RAW data file. After additional tests, logical acquisitions of the Samsung were taken as full physical images were not necessary. For the iPhone, it is important to note that while the iPhone was jailbroken, Magnet Acquire only supported the acquisition of logical images [28].

4 Experimental Results

To analyze all experimental results and extract relevant artifacts, different tools shown in Table 1 were utilized along with manual analysis. The next subsections discuss all artifacts of interest identified per device. Appendix B, Table 2, references those significant artifacts and their data directory paths.

[9] https://www.magnetforensics.com/resources/magnet-acquire/.

4.1 Major Artifacts Found - Apple CarPlay

CarPlay Data Structure: To identify artifacts related to the Apple CarPlay feature, understanding how data is stored on an iOS device is critical. Apple CarPlay is part of Apple's SpringBoard framework that manages the screen of an iOS (or macOS) device [13]. Thus, unlike the Android Auto application, the hands-free features of CarPlay are built in the main settings of the phone rather than being compiled in a sole separate application. Therefore, the settings files associated with CarPlay are spread out throughout the phone and not compartmentalized like a normal application. If the iPhone is connected to more than one head unit that supports Apple CarPlay, the feature names the settings' files with a Globally Unique Identifier (GUID) associated with each head unit that can later be used to correlate data found in the settings' files [16]. Throughout our research, it was determined that data pertaining to other applications or features used while implementing CarPlay was not found within the CarPlay settings. CarPlay simply serves as a projection feature to assist users in viewing and accessing every day items such as music, messages, and phone calls and any other application with support for CarPlay. However, we realized there were still relevant pieces of information that are recoverable that can help investigators gain a glimpse into what was being displayed or used while an individual was driving. The following subsections denote the relevant files where this information can be found.

cache.plist: This artifact contained data pertaining to timestamps associated with core phone settings and other location calibration information unrelated to CarPlay (Appendix B, Table 2, File ID 1). The most important data i n this artifact is the "Last Vehicle Connection" entry which provides the connect and disconnect timestamps in Cocoa Core Data format for the last time the device was connected to a vehicle. This entry also notes the name of the vehicle the iPhone was last connected to. The latter is relevant in an investigation to confirm a timeline of events and that in fact the user's phone was connected to the vehicle.

com.apple.carplay.plist: This pList file contained information about the vehicle pairings associated with the car (Appendix B, Table 2, File ID 2). Since the device was connected to two car audio head units, one custom built and one with an actual car, two GUIDs were found, along with strings that identify the model of the units: iLX-W650 and Subaru respectively (Fig. 3). This information uniquely identifies and confirms pairings to different devices. Additionally, the file contains "carPlayProtocols" which were found to be additional applications that come as part of the standard head unit's settings. In the case of the Subaru, the head unit comes with support for Aha, a road map software, Pandora, and Subaru's global infotainment application.

com.apple.celestial.plist: This pList file contained settings for many different features of the phone (Appendix B, Table 2, File ID 3). Among these

pairings	Dictionary	(2 items)
ED2504E9-67E5-4608-9937-FB4E43DA1056	Dictionary	(2 items)
carPlayProtocols	Array	(0 items)
name	String	iLX-W650
F23EBC12-6249-45EC-8396-8955719E84AD	Dictionary	(2 items)
carPlayProtocols	Array	(3 items)
name	String	Subaru

Fig. 3. Head unit pairings

features are volume settings for different connections, information about the camera, and IDs for the applications on the display when CarPlay was disconnected. Data of interest in this file includes, the strings "nowPlayingAppWasPlayingUponCarPlayDisconnect" and "nowPlayingAppDisplayIDUponCarPlayDisconnect" which denote the App ID associated with the last app running on CarPlay (Fig. 4). It is interesting to note that for some tests, the string "nowPlayingApp" noted Apple's "Music" app was playing although this app was not used during the tests. Conclusions as to why this occurred were not formally reached but in some instances the values for the "nowPlayingApp" matched that of the strings related to the application last being used upon disconnect from the car. Additionally, there was a "CarAudioOutput-"Car ID"" entries with a unique identifier for each car which denoted what appeared to be the volume level for when the phone is connected to the car.

Key	Type	Value
Root	Dictionary	(8 items)
mobileAssetUpdateTimes	Dictionary	(2 items)
volumeMultiplier	Dictionary	(1 items)
nowPlayingAppDisplayID	String	com.apple.Music
nowPlayingAppWasPlayingUponCarPlayDisconnect	Boolean	true
CaptureSourceInfo	Dictionary	(5 items)
nowPlayingAppDisplayIDUponCarPlayDisconnect	String	com.spotify.client
volumes	Dictionary	(6 items)
CarAudioOutput~28:56:C1:3B:5A:CB-Audio-AudioMain-...	Dictionary	(1 items)
CarAudioOutput~2C:1C:2E:AA:86:46-Audio-AudioMain-...	Dictionary	(1 items)

Fig. 4. CarPlay com.apple.celestial.plist

com.apple.springboard.plist: This file contained general and user settings for the SpringBoard application which manages the home screen for iOS devices [17] (Appendix B, Table 2, File ID 4). While many of the entries in this file are not relevant to CarPlay, there is an entry which provides potential insight into

the most recent activities. The heading "CarDisplayRecentlyUsedApps" (Fig. 5) shows values for the package names of the applications last used on the display. In this example, while final tests were performed, calls were placed, music was played, and map directions were given.

CarDisplayRecentlyUsedApps	Array	(3 items)
item 0	String	com.apple.mobilephone
item 1	String	com.spotify.client
item 2	String	com.apple.Maps
CarDisplayStartApp	String	com.apple.springboard

Fig. 5. CarPlay recently used applications

"Car GUID"-CarPlay[Desired/Display]IconState.plist: These pList files contained data pertaining to the Icon Layout on the CarPlay display (Appendix B, Table 2, File IDs 5, 6). Each car had its own unique application layout for the applications installed on the phone. A user can change this by visiting the CarPlay settings located within the General settings of the iPhone (Figs. 6 and 7).

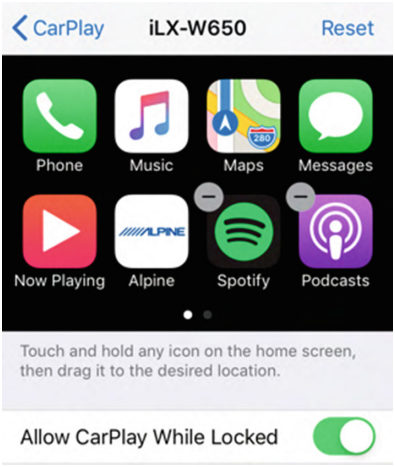


Fig. 6. CarPlay application settings

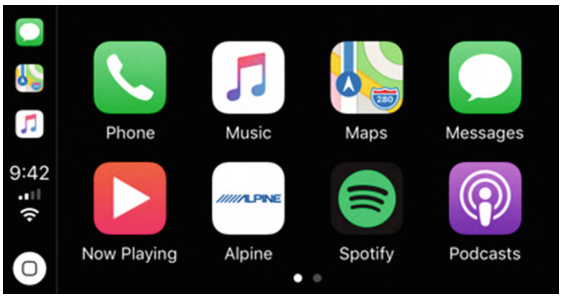


Fig. 7. CarPlay headunit apps

carplay_connect_timestamp and carplay_success_timestamp: It is important to note that these artifacts did not contain any data but their name and timestamps may be relevant to an investigation (Appendix B, Table 2, File IDs 7, 8). These files were found in the Assistant directory, associated with Siri, and the timestamps are associated with actions completed between CarPlay and Siri. The "carplay_connect_timestamp" is the timestamp for the last initial connection to the vehicle (i.e. turning the vehicle on at the start of a drive and connecting the phone). The "carplay_success_timestamp" is associated with the last time Siri was used while using CarPlay. This was verified after comparing testing notes for asking Siri for directions to the timestamp associated with the file.

4.2 Major Artifacts Found - Android Auto

Android Auto Data Structure: The structure of the data files stored by Android Auto follows that of a normal application on the phone. Unlike Apple CarPlay, Android Auto is an application that can be downloaded from the Google Play Store. The important data files were found within the "com.google.android.projection.gearhead" folder within the /data/data directory and are broken into XML and SQLite databases. The most important artifacts found are discussed in the following sub-sections.

carservicedata.db: This database (Appendix B, Table 2, File ID 9) included information verifying the stereo head units the mobile device has connected to. It had two tables which separated the "allowedcars" and "rejectedcars". Both tables contained data about the manufacturer, the vehicle ID, Bluetooth addresses and allowed connections. It is interesting to note that the first twelve characters of the Subaru's vehicle ID are the same used for its Bluetooth address. The "connectiontime" value is the timestamp associated with the first time the phone was connected to the car within a given driving session (i.e., initially turning on vehicle and connecting to the car). To approximate the accuracy of that timestamp, the "app_state_shared_preferences.xml" file (Appendix B, Table 2,

File ID 12) stored a "pref_lastrun" value that depicted the date and time the Android Auto application was last run. The timestamp in the carservicedata.db will always be a few seconds ahead of the other as the application is opened first before connecting to Bluetooth.

Additionally, the database made a distinction if the head unit was "After Market", which refers to the custom built head unit, or associated with a brand model such as Subaru. Finally, the table had four more columns pertaining to WiFi information such as the WiFi Service Set Identifier (SSID), Basic service set identifier (BSSID), password and security for vehicles that have WiFi. Note, the latter was not tested.

common_user_settings.xml: This XML file (Appendix B, Table 2, File ID 10 and Fig. 8) contained information about the preferred settings for each head unit the device has connected to. For instance, enabled messaging notifications, visual preview of messages, auto reply messages while driving and more. Furthermore, for every vehicle, there is a section identified by the Bluetooth Address pertaining to each vehicle and a "USB" value denoting the device was also connected via a USB cable. It is interesting to note that for the Subaru, one could assign a custom name to the vehicle's Bluetooth connection. In this research, the Subaru's connection name was "Bluebaru."

```
<string name="key_settings_carmode_screen_on">SCREEN_ON_POLICY_ALWAYS_ON</string>
<boolean name="key_settings_autolaunch_delay_proximity" value="true"/>
<string name="key_settings_notifications_auto_reply_message">I'm driving right now</string>
<set name="AndroidAutoBluetooth_28:56:C1:3B:5A:CB">
  <string>USB</string>
```

Fig. 8. Android auto common user settings

default_app.xml: This file (Appendix B, Table 2, File ID 11) provided a method to verify package names of the default applications that were displayed on the Android Auto Interface of the stereo head unit (Fig. 9).

auto_launch_prompt.xml: This artifact (Appendix B, Table 2, File ID 12) is associated with the application's auto launch feature. It contains the Bluetooth Addresses to the vehicles the device has ever connected to.

carservice.xml: This file (Appendix B, Table 2, File ID 13) contained car_module_feature_set values which define settings and information for the connections to the car. It is important to note when the Samsung device was first connected to the custom built stereo, this file contained minimal settings. However, after connecting to the Subaru vehicle's head unit, this file supported additional settings such as "car night mode" and enabling/disabling "connection to known cars only" among other settings.

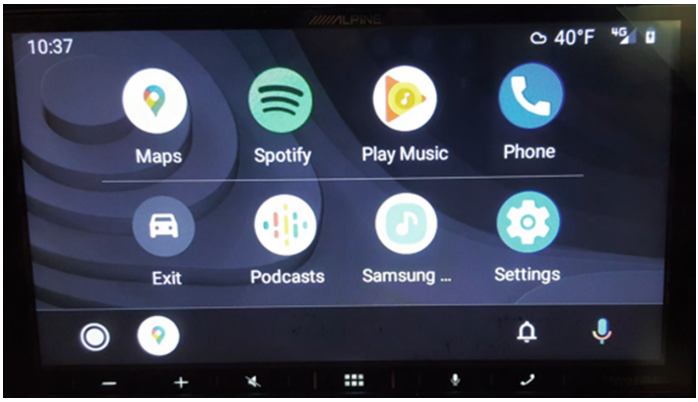


Fig. 9. Default displayed applications

5 Tool Creation and Usage

Overview: The purpose of this tool is to assist the forensic community in automatically identifying relevant artifacts within Apple and Android smartphone forensic disk images. The artifacts created while using these applications paired to the vehicle, could possibly aid in determining causes of different incidents, including car crashes.

A Python tool was constructed using a wordlist to search through the forensic tar images of Apple and Android smartphones. Note, that only tar formats are supported now, however, support for other types may be added in the future. Based on our manual analysis and identification of relevant forensic artifacts and their storage locations on the devices, default wordlists were created. The "words" used in this case could be full file paths and their file names, solely artifact file names, or other useful keywords. For a complete list, see Appendix B, Table 2. A default wordlist containing all relevant and important paths for an investigation is provided if the user does not add a custom one themselves. For the default wordlist, once the critical files are extracted, they are parsed to return specific data in the form of an HTML report. Note, if the default wordlist is edited to add more files, these files will be extracted but will not be included in the report. This also applies to the custom wordlist option. The high level algorithm for this tool is shown in Algorithm 1.

For Apple devices, wav, pList and SQLite file types were the focus within the report. The wav files store a user's voice commands when using Apple's Siri. The pList files store information about the Apple device settings and the vehicle's head unit identifiers. The SQLite files hold information on recent phone calls and text messages. For the Android devices Binarypb, XML, and SQLite file types were focused on to extract and parse. The Binarypb files store a user's voice commands using Android Assistant. The XML and SQLite files store information about the Androids device settings and the vehicle's head unit identifiers. The SQLite databases also store information on recent phone calls and text messages.

Algorithm 1. High-level Automation Algorithm

Requirements: Python3 and a TAR image of a device.
Input: iPhone or Android Mobile tar image, (optional) inputFile with keywords
Output: Apple Carplay & Android Auto artifacts and HTML Forensics Report

```
Select image type:
if Apple option then
    iPhone_artifact();
else if Android option then
    Android_artifact();

Specified image analysis
if Wordlist option not set then
    initial_hashing();                          ▷ Obtain hashes of images for verification
    search_archive(default wordlist);           ▷ Searches archive for default values
    extract_found();                            ▷ Extracts files located during search to folder
    analyze_files();                            ▷ Parses data
    check_hashing();                            ▷ Verification of hashes
    generate_report();                          ▷ Generates HTML report detailing findings

else
    initial_hashing();                          ▷ Obtain hashes of images for verification
    search_archive(user wordlist);              ▷ Searches archive for user values
    extract_found();                            ▷ Extracts files located during search to folder
    check_hashing();                            ▷ Verification of hashes
    generate_report();                          ▷ Generates HTML report header
```

Usage: The tool is portable and designed to be used with the command line terminal (Fig. 10). The user may want to use either an Apple or an Android forensic disk image and there are two flags to denote these options. There is also an option to select a custom user created wordlist. Since the parsing/grouping analysis is specific to the default wordlist, if a user chooses their own wordlist an examination will not be done to produce an actual HTML report. Instead, the tool will only extract files or file paths that match words in the wordlist. When the tool is run, the case number and examiners name can be entered.

```
Usage:   Tool.py [options] <inputfile>
Example: Tool.py -a Apple.tar -w wordlist.txt
Options:
         -h, --help
         -a            apple image tar
         -b            android image tar
         -w            user created wordlist (not required for default tool use)
```

Fig. 10. Tool's usage help menu

Output: The tool will organize the recovered files into a generated folder. The first part of the name being the timestamp the program ran and the second part

being either "Apple" or "Android" depending on the type of image being used. The results of the analysis will output in table form. This will be displayed to the user on the console as well as an HTML report. Along with the data the output will contain the timestamp taken, examiner's name, and case number. Both the md5 and sha256 hashes will be taken of the image before and after its analysis to ensure changes were not made during processing. It is important to note that this report will only be generated when the default wordlists are used. Figures 11 and 12 in Appendix A demonstrate sample generated output from cases that have Apple and Android forensic images as input.

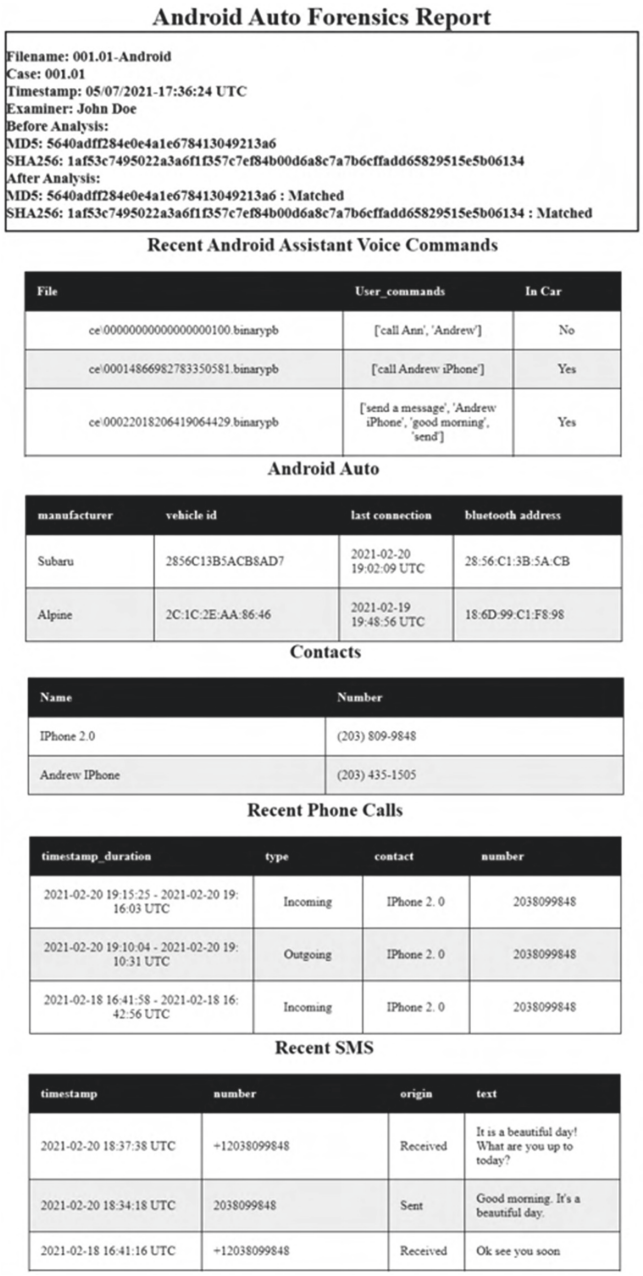
6 Conclusion and Future Work

The world is heavily reliant on vehicles and mobile phones now more than ever. Understanding the interaction between phones and vehicles can be important to investigators. While this research confirmed that Android Auto and Apple CarPlay only serve as projection methods for the applications that offer support for these features, we concluded there is still relevant information that can be acquired for use in crash investigations. By determining any vehicles that a device has been connected to, investigators can link suspects to cars left at crime scenes or to other investigations. Additionally, investigators can determine possible areas of distraction when examining the default home application settings for the Apple CarPlay or Android Auto interfaces. These areas of data provide a greater insight into how the phone was connected to the vehicle and the way in which a user interacts with it. The tool presented offers a fast and compact solution for obtaining the relevant information such as vehicles the device has been connected to, timestamps, text message and phone call histories, and other relevant information investigators may need.

Future work must be focused on maintaining the tool to adapt to how the applications' files change over time and as new features are added to the applications and the vehicle head units. Additionally, some examiners may not have imaging software that supports tar outputs, thus, support for additional forensic image file types would increase the usability of the tool.

Acknowledgement. This material is based upon work supported by the National Science Foundation under Grant Numbers 1900210 and 1921813. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Additionally, the authors would like to thank Mark Morton for his help in designing and building the stereo head unit.

A Appendix - Sample Tool Reports



Android Auto Forensics Report

Filename: 001.01-Android
Case: 001.01
Timestamp: 05/07/2021-17:36:24 UTC
Examiner: John Doe
Before Analysis:
MD5: 5640adff284e0e4a1e678413049213a6
SHA256: 1af53c7495022a3a6f1f357c7ef84b00d6a8c7a7b6cffadd65829515e5b06134
After Analysis:
MD5: 5640adff284e0e4a1e678413049213a6 : Matched
SHA256: 1af53c7495022a3a6f1f357c7ef84b00d6a8c7a7b6cffadd65829515e5b06134 : Matched

Recent Android Assistant Voice Commands

File	User_commands	In Car
ce\0000000000000000100.binarypb	['call Ann', 'Andrew']	No
ce\0001486698278335058l.binarypb	['call Andrew iPhone']	Yes
ce\0002201820641906442 9.binarypb	['send a message', 'Andrew iPhone', 'good morning', 'send']	Yes

Android Auto

manufacturer	vehicle id	last connection	bluetooth address
Subaru	2856C13B5ACB8AD7	2021-02-20 19:02:09 UTC	28:56:C1:3B:5A:CB
Alpine	2C:1C:2E:AA:86:46	2021-02-19 19:48:56 UTC	18:6D:99:C1:F8:98

Contacts

Name	Number
IPhone 2.0	(203) 809-9848
Andrew IPhone	(203) 435-1505

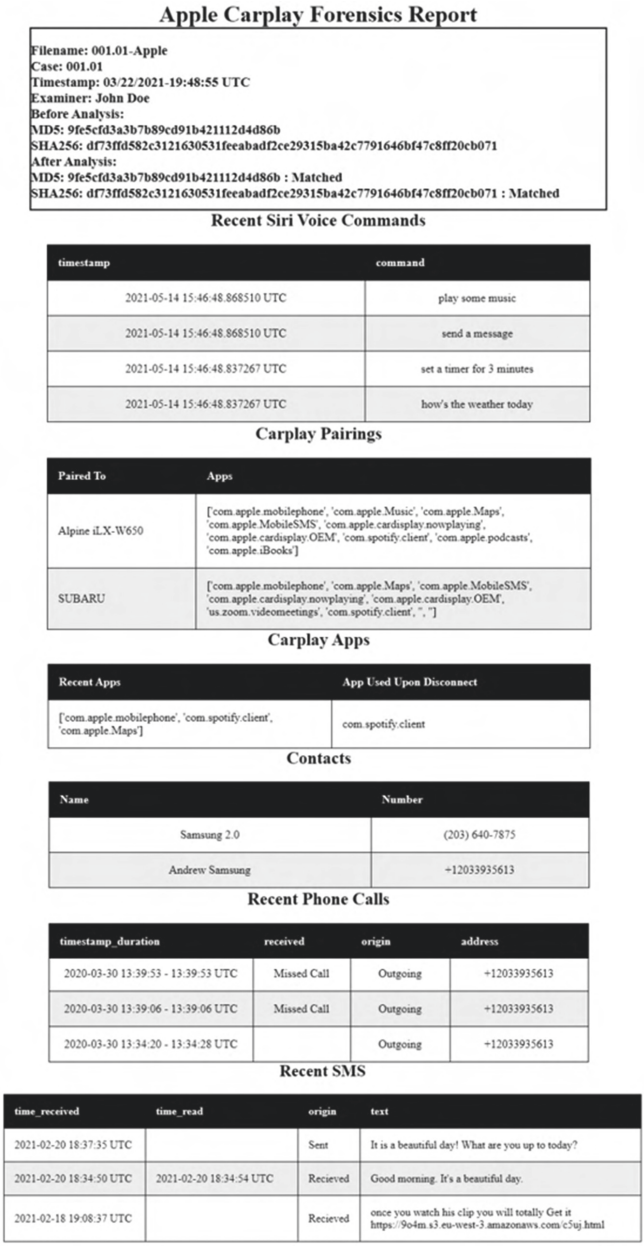
Recent Phone Calls

timestamp_duration	type	contact	number
2021-02-20 19:15:25 - 2021-02-20 19: 16:03 UTC	Incoming	IPhone 2. 0	2038099848
2021-02-20 19:10:04 - 2021-02-20 19: 10:31 UTC	Outgoing	IPhone 2. 0	2038099848
2021-02-18 16:41:58 - 2021-02-18 16: 42:56 UTC	Incoming	IPhone 2. 0	2038099848

Recent SMS

timestamp	number	origin	text
2021-02-20 18:37:38 UTC	+12038099848	Received	It is a beautiful day! What are you up to today?
2021-02-20 18:34:18 UTC	2038099848	Sent	Good morning. It's a beautiful day.
2021-02-18 16:41:16 UTC	+12038099848	Received	Ok see you soon

Fig. 11. Sample android auto output



Apple Carplay Forensics Report

Filename: 001.01-Apple
Case: 001.01
Timestamp: 03/22/2021-19:48:55 UTC
Examiner: John Doe
Before Analysis:
MD5: 9fe5cfd3a3b7b89cd91b421112d4d86b
SHA256: df73ffd582c3121630531feeabadf2ce29315ba42c7791646bf47c8ff20cb071
After Analysis:
MD5: 9fe5cfd3a3b7b89cd91b421112d4d86b : Matched
SHA256: df73ffd582c3121630531feeabadf2ce29315ba42c7791646bf47c8ff20cb071 : Matched

Recent Siri Voice Commands

timestamp	command
2021-05-14 15:46:48.868510 UTC	play some music
2021-05-14 15:46:48.868510 UTC	send a message
2021-05-14 15:46:48.837267 UTC	set a timer for 3 minutes
2021-05-14 15:46:48.837267 UTC	how's the weather today

Carplay Pairings

Paired To	Apps
Alpine iLX-W650	['com.apple.mobilephone', 'com.apple.Music', 'com.apple.Maps', 'com.apple.MobileSMS', 'com.apple.cardisplay.nowplaying', 'com.apple.cardisplay.OEM', 'com.spotify.client', 'com.apple.podcasts', 'com.apple.iBooks']
SUBARU	['com.apple.mobilephone', 'com.apple.Maps', 'com.apple.MobileSMS', 'com.apple.cardisplay.nowplaying', 'com.apple.cardisplay.OEM', 'us.zoom.videomeetings', 'com.spotify.client', '', '']

Carplay Apps

Recent Apps	App Used Upon Disconnect
['com.apple.mobilephone', 'com.spotify.client', 'com.apple.Maps']	com.spotify.client

Contacts

Name	Number
Samsung 2.0	(203) 640-7875
Andrew Samsung	+12033935613

Recent Phone Calls

timestamp_duration	received	origin	address
2020-03-30 13:39:53 - 13:39:53 UTC	Missed Call	Outgoing	+12033935613
2020-03-30 13:39:06 - 13:39:06 UTC	Missed Call	Outgoing	+12033935613
2020-03-30 13:34:20 - 13:34:28 UTC		Outgoing	+12033935613

Recent SMS

time_received	time_read	origin	text
2021-02-20 18:37:35 UTC		Sent	It is a beautiful day! What are you up to today?
2021-02-20 18:34:50 UTC	2021-02-20 18:34:54 UTC	Recieved	Good morning. It's a beautiful day.
2021-02-18 19:08:37 UTC		Recieved	once you watch his clip you will totally Get it https://9o4m.s3.eu-west-3.amazonaws.com/c5uj.html

Fig. 12. Sample apple CarPlay output

B Appendix - Table 2 - Relevant Artifacts

Table 2. Important data path directories and files found in disk

File ID	Path	OS	Description
1	/private/var/root/Library/Caches/locationd/cache.plist	iOS	Last Vehicle Connection Timestamps & vehicle name
2	XXX/Preferences/com.apple.carplay.plist	iOS	CarPlay Pairings & Protocols
3	XXX/Preferences/com.apple.celestial.plist	iOS	Volume Settings, Camera, Last App Displayed on CarPlay
4	XXX/Preferences/com.apple.springboard.plist	iOS	SpringBoard Settings & CarPlay Recently Used Apps
5	XXX/SpringBoard/"Car GUID"-CarPlayDesiredIconState.plist	iOS	User Display Icons Layout
6	XXX/SpringBoard/"Car GUID"-CarPlayDisplayIconState.plist	iOS	User Display Icons
7	XXX/Assistant/carplay_connect_timestamp	iOS	No Data but interesting
8	XXX/Assistant/carplay_success_timestamp	iOS	No Data but interesting
9	***/databases/carservicedata.db	Android	Accepted/Rejected Car Connections
10	***/shared_prefs/common_user_settings.xml	Android	User Settings
11	***shared_prefs/default_app.xml	Android	Default Screen App Layouts
12	***shared_prefs/auto_launch_prompt.xml	Android	Auto Launch Prompt Count for Vehicle Connections
13	***shared_prefs/carservice.xml	Android	Car Feature Settings

Base iOS Path: XXX = /private/var/mobile/Library — Base Android Path: *** = vol_vol20/data/com.google.android.projection.gearhead

References

1. National Highway Traffic Safety Administration: Overview of motor vehicle crashes in 2019. Traffic Safety Facts Research Note. Report No. DOT HS 813 060, December 2020. https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/813060
2. Al Mutawa, N., Baggili, I., Marrington, A.: Forensic analysis of social networking applications on mobile devices. Digit. Investig. **9**, S24–S33 (2012)
3. Android: Android auto compatibility. https://www.android.com/auto/compatibility-/#compatibility-vehicles
4. Anglano, C., Canonico, M., Guazzone, M.: Forensic analysis of telegram messenger on android smartphones, September 2017. https://www.sciencedirect.com/science/article/abs/pii/S-1742287617301767
5. Bader, M., Baggili, I.: iphone 3GS forensics: logical analysis using apple itunes backup utility (2010)
6. Casey, P., Baggili, I., Yarramreddy, A.: Immersive virtual reality attacks and the human joystick. IEEE Trans. Dependable Secure Comput. 1 (2019)
7. CDC: Distracted driving, March 2021. https://www.cdc.gov/transportationsafety/distracted_driving/index.html
8. Clark, D.R., Meffert, C., Baggili, I., Breitinger, F.: Drop (drone open source parser) your drone: forensic analysis of the DJI phantom III. Digit. Investig. **22**, S3–S14 (2017)
9. Statista Research Department: Worldwide - connected car shipments, August 2021. https://www.statista.com/statistics/743400/estimated-connected-car-shipments-globally/
10. Dorai, G., Houshmand, S., Baggili, I.: I know what you did last summer: your smart home internet of things and your iphone forensically ratting you out. In: Proceedings of the 13th International Conference on Availability, Reliability and Security, ARES 2018. Association for Computing Machinery, New York (2018). https://doi.org/10.1145/3230833.3232814
11. Ebbers, S., Ising, F., Saatjohann, C., Schinzel, S.: Grand theft app: digital forensics of vehicle assistant apps. CoRR abs/2106.04974 (2021). https://arxiv.org/abs/2106.04974
12. Edwards, S., Mahalik, H.: They see us rollin'; they hatin': Forensics of iOS carplay and android auto. SANS DIFR Summit (2019). https://www.youtube.com/watch?v=IGhXsfZXL6g
13. iPhone FAQ, T.: What is the iphone iOS springboard? April 2015. https://www.iphonefaq.org/archives/971473
14. Graham, J.: Apple iphone again best tech seller of the year, thanks to the working-from-home trend, December 2020. https://www.usatoday.com/story/tech/2020/12/28/apple-iphone-best-selling-tech-device-year-work-home-helped/4048111001/
15. Grajeda, C., Sanchez, L., Baggili, I., Clark, D., Breitinger, F.: Experience constructing the artifact genome project (AGP): managing the domain's knowledge one artifact at a time. Digit. Investig. **26**, S47–S58 (2018)
16. guid.one: what is a GUID? http://guid.one/guid
17. Hamilton, D.: Corrupt iCloud data causes iOS springboard home screen crash (with fix!), April 2013. https://www.macobserver.com/tmo/article/corrupt-icloud-data-can-cause-ios-springboard-home-screen-crash

18. Hassenfeldt, C., Baig, S., Baggili, I., Zhang, X.: Map my murder. In: Proceedings of the 14th International Conference on Availability, Reliability and Security - ARES 2019 (2019). https://doi.org/10.1145/3339252.3340515
19. Hickman, J.: Ka-chow!!! Driving android auto (2019). https://thebinaryhick.blog/2019/01/30/ka-chow-driving-android-auto/
20. Hickman, J.: Ka-chow!!! Driving android auto. DFIR Review (2019). https://dfir.pubpub.org/pub/716tlra7/release/2
21. Hickman, J.: Ridin' with apple carplay, May 2019. https://thebinaryhick.blog/2019/05/08/ridin-with-apple-carplay/
22. Husain, M.I., Baggili, I., Sridhar, R.: A simple cost-effective framework for iphone forensic analysis. In: Baggili, I. (ed.) ICDF2C 2010. LNICST, vol. 53, pp. 27–37. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19513-6_3
23. Iqbal, A., Alobaidli, H., Marrington, A., Baggili, I.: Amazon kindle fire HD forensics. In: Gladyshev, P., Marrington, A., Baggili, I. (eds.) ICDF2C 2013. LNICSSITE, vol. 132, pp. 39–50. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-14289-0_4
24. Jacobs, D., Choo, K.K.R., Kechadi, M.T., Le-Khac, N.A.: Volkswagen car entertainment system forensics. In: 2017 IEEE Trustcom/BigDataSE/ICESS (2017). https://doi.org/10.1109/trustcom/bigdatase/icess.2017.302
25. Karpisek, F., Baggili, I., Breitinger, F.: Whatsapp network forensics: decrypting and understanding the whatsapp call signaling messages (2015)
26. Lacroix, J.: Vehicular infotainment forensics: collecting data and putting it into perspective (2017)
27. Lacroix, J., El-Khatib, K., Akalu, R.: Vehicular digital forensics: what does my vehicle know about me? In: DIVANet 2016: Proceedings of the 6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications, pp. 59–66 (2016)
28. Magnet Forensics: Magnet acquire (2020). https://www.magnetforensics.com/resources/magnet-acquire/
29. Mahalik, H.: How android bluetooth connections can determine if a driver had their hands on the wheel during an accident. DFIR Review (2020). https://dfir.pubpub.org/pub/6ysxvhvc/release/1
30. Mahr, A., Cichon, M., Mateo, S., Grajeda, C., Baggili, I.: Zooming into the pandemic! a forensic analysis of the zoom application. Forensic Sci. Int. Digit. Investig. **36**, 301107 (2021)
31. Mays, K.: Wireless apple carplay and android auto: Where are they now?: news from cars.com, December 2020. https://www.cars.com/articles/wireless-apple-carplay-and-android-auto-where-are-they-now-407297/
32. Moore, J., Baggili, I., Breitinger, F.: Find me if you can: mobile GPS mapping applications forensic analysis & SNAVP the open source, modular, extensible parser. J. Digit. Forensics Secur. Law **12**(1), 7 (2017)
33. Moore, J., Baggili, I., Marrington, A., Rodrigues, A.: Preliminary forensic analysis of the xbox one. Digit. Investig. **11**, S57–S65 (2014)
34. Statista: Samsung smartphone sales 2020, March 2021. https://www.statista.com/statistics/299144/samsung-smartphone-shipments-worldwide/
35. Walnycky, D., Baggili, I., Marrington, A., Moore, J., Breitinger, F.: Network and device forensic analysis of android social-messaging applications, August 2015. https://www.sciencedirect.com/science/article/pii/S1742-287615000547
36. Yarramreddy, A., Gromkowski, P., Baggili, I.: Forensic analysis of immersive virtual reality social applications: a primary account. In: 2018 IEEE Security and Privacy Workshops (SPW), pp. 186–196 (2018). https://doi.org/10.1109/SPW.2018.00034

37. Zhang, S., Wang, L.: Forensic analysis of social networking application on iOS devices. In: Sixth International Conference on Machine Vision (ICMV 2013), vol. 9067, p. 906715. International Society for Optics and Photonics (2013)
38. Zhang, X., Baggili, I., Breitinger, F.: Breaking into the vault: privacy, security and forensic analysis of android vault applications, August 2017. https://www.sciencedirect.com/science/article/pii/S01674-04817301529